Федеральное государственное автономное образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

Направление подготовки 09.03.04 «Программная инженерия» –

Системное и прикладное программное обеспечение

**Отчёт**

**По лабораторной работе №4**

по дисциплине «Базы данных»

**Вариант: 3569**

Выполнил:

студент 1 курса

Батманов Даниил Евгеньевич

Группа: Р3107

Приняла:

Бострикова Дарья Константиновна

Отчёт принят «\_\_»\_\_\_\_\_2023г.

Оценка: \_\_\_\_\_\_\_\_\_\_\_

г. Санкт-Петербург, 2023

**Оглавление**

[Задание 3](#_Toc129812876)

[Основные этапы работы 5](#_Toc129812877)

[Заключение 11](#_Toc129812878)

# 

# 

# 

# Задание

Составить запросы на языке SQL (пункты 1-2).

Для каждого запроса предложить индексы, добавление которых уменьшит время выполнения запроса (указать таблицы/атрибуты, для которых нужно добавить индексы, написать тип индекса; объяснить, почему добавление индекса будет полезным для данного запроса).

Для запросов 1-2 необходимо составить возможные планы выполнения запросов. Планы составляются на основании предположения, что в таблицах отсутствуют индексы. Из составленных планов необходимо выбрать оптимальный и объяснить свой выбор.  
Изменятся ли планы при добавлении индекса и как?

Для запросов 1-2 необходимо добавить в отчет вывод команды EXPLAIN ANALYZE [запрос]

Подробные ответы на все вышеперечисленные вопросы должны присутствовать в отчете (планы выполнения запросов должны быть нарисованы, ответы на вопросы - представлены в текстовом виде).

1. Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:  
   Таблицы: Н\_ЛЮДИ, Н\_СЕССИЯ.  
   Вывести атрибуты: Н\_ЛЮДИ.ФАМИЛИЯ, Н\_СЕССИЯ.ДАТА.  
   Фильтры (AND):   
   a) Н\_ЛЮДИ.ОТЧЕСТВО = Александрович.  
   b) Н\_СЕССИЯ.ЧЛВК\_ИД < 106059.  
   c) Н\_СЕССИЯ.ЧЛВК\_ИД = 126631.  
   Вид соединения: INNER JOIN.
2. Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:  
   Таблицы: Н\_ЛЮДИ, Н\_ОБУЧЕНИЯ, Н\_УЧЕНИКИ.  
   Вывести атрибуты: Н\_ЛЮДИ.ОТЧЕСТВО, Н\_ОБУЧЕНИЯ.НЗК, Н\_УЧЕНИКИ.НАЧАЛО.  
   Фильтры: (AND)  
   a) Н\_ЛЮДИ.ФАМИЛИЯ < Ёлкин.  
   b) Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД > 163276.  
   Вид соединения: LEFT JOIN.

# Основные этапы работы

***Реализация первого запроса***

*Задание:*

Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:  
Таблицы: Н\_ЛЮДИ, Н\_СЕССИЯ.  
Вывести атрибуты: Н\_ЛЮДИ.ФАМИЛИЯ, Н\_СЕССИЯ.ДАТА.  
Фильтры (AND):   
a) Н\_ЛЮДИ.ОТЧЕСТВО = Александрович.  
b) Н\_СЕССИЯ.ЧЛВК\_ИД < 106059.  
c) Н\_СЕССИЯ.ЧЛВК\_ИД = 126631.  
Вид соединения: INNER JOIN.

*Запрос:*

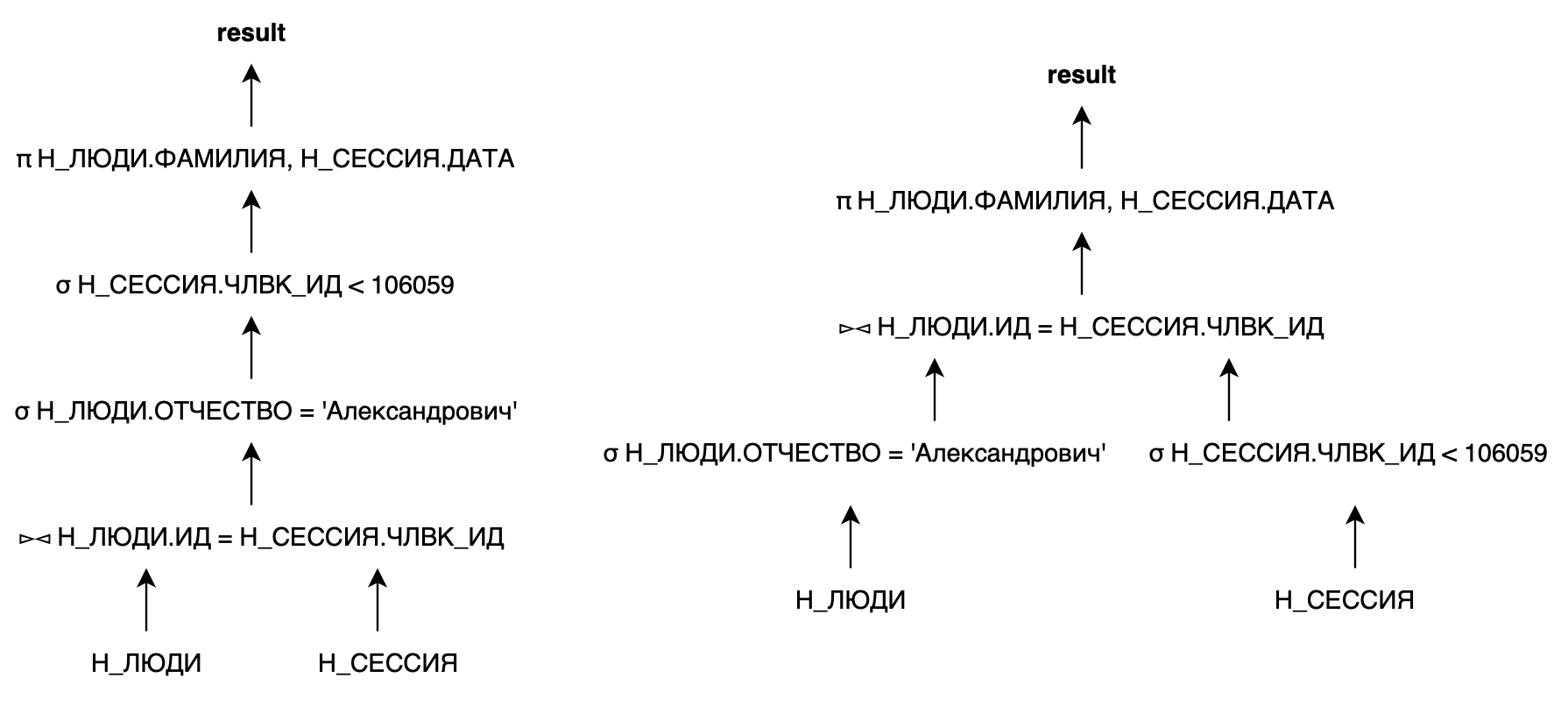
SELECT Н\_ЛЮДИ.ФАМИЛИЯ, Н\_СЕССИЯ.ДАТА

FROM Н\_ЛЮДИ

INNER JOIN Н\_СЕССИЯ ON Н\_ЛЮДИ.ИД = Н\_СЕССИЯ.ЧЛВК\_ИД

WHERE (Н\_ЛЮДИ.ОТЧЕСТВО = 'Александрович' AND Н\_СЕССИЯ.ЧЛВК\_ИД < 106059);

*Планы выполнения:*



Во втором плане происходит объединение только нужной выборки, а не всех сущностей. Размер промежуточных данных меньше, значит этот план является оптимальным.

*Индексы:*

CREATE INDEX ON "Н\_ЛЮДИ" USING HASH ("ИД");

CREATE INDEX ON "Н\_ЛЮДИ" USING BTREE ("ОТЧЕСТВО");

CREATE INDEX ON "Н\_СЕССИЯ" USING BTREE ("ЧЛВК\_ИД");

Выборка происходит с использованием операторов сравнения, поэтому оптимально использование BTREE. Соединение таблиц с использованием ИД – оптимально использование HASH.

*Explain Analyze:*

EXPLAIN ANALYZE

SELECT Н\_ЛЮДИ.ОТЧЕСТВО, Н\_СЕССИЯ.ДАТА

FROM Н\_ЛЮДИ

INNER JOIN Н\_СЕССИЯ ON Н\_ЛЮДИ.ИД = Н\_СЕССИЯ.ЧЛВК\_ИД

WHERE (Н\_ЛЮДИ.ОТЧЕСТВО = 'Александрович' AND Н\_СЕССИЯ.ЧЛВК\_ИД < 106059)

Nested Loop (cost=0.29..271.57 rows=231 width=28) (actual time=0.054..1.864 rows=248 loops=1)

-> Seq Scan on "Н\_СЕССИЯ" (cost=0.00..117.90 rows=2726 width=12) (actual time=0.012..0.698 rows=2728 loops=1)

Filter: ("ЧЛВК\_ИД" < 106059)

Rows Removed by Filter: 1024

-> Memoize (cost=0.29..0.48 rows=1 width=24) (actual time=0.000..0.000 rows=0 loops=2728)

Cache Key: "Н\_СЕССИЯ"."ЧЛВК\_ИД"

Cache Mode: logical

Hits: 2624 Misses: 104 Evictions: 0 Overflows: 0 Memory Usage: 8kB

-> Index Scan using "ЧЛВК\_PK" on "Н\_ЛЮДИ" (cost=0.28..0.47 rows=1 width=24) (actual time=0.002..0.002 rows=0 loops=104)

Index Cond: ("ИД" = "Н\_СЕССИЯ"."ЧЛВК\_ИД")

Filter: (("ОТЧЕСТВО")::text = 'Александрович'::text)

Rows Removed by Filter: 1

Planning Time: 1.132 ms

Execution Time: 1.966 ms

***Реализация второго запроса***

*Задание:*

Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:  
Таблицы: Н\_ЛЮДИ, Н\_ОБУЧЕНИЯ, Н\_УЧЕНИКИ.  
Вывести атрибуты: Н\_ЛЮДИ.ОТЧЕСТВО, Н\_ОБУЧЕНИЯ.НЗК, Н\_УЧЕНИКИ.НАЧАЛО.  
Фильтры: (AND)  
a) Н\_ЛЮДИ.ФАМИЛИЯ < Ёлкин.  
b) Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД > 163276.  
Вид соединения: LEFT JOIN.

*Запрос:*

SELECT Н\_ЛЮДИ.ОТЧЕСТВО, Н\_ОБУЧЕНИЯ.НЗК, Н\_УЧЕНИКИ.НАЧАЛО

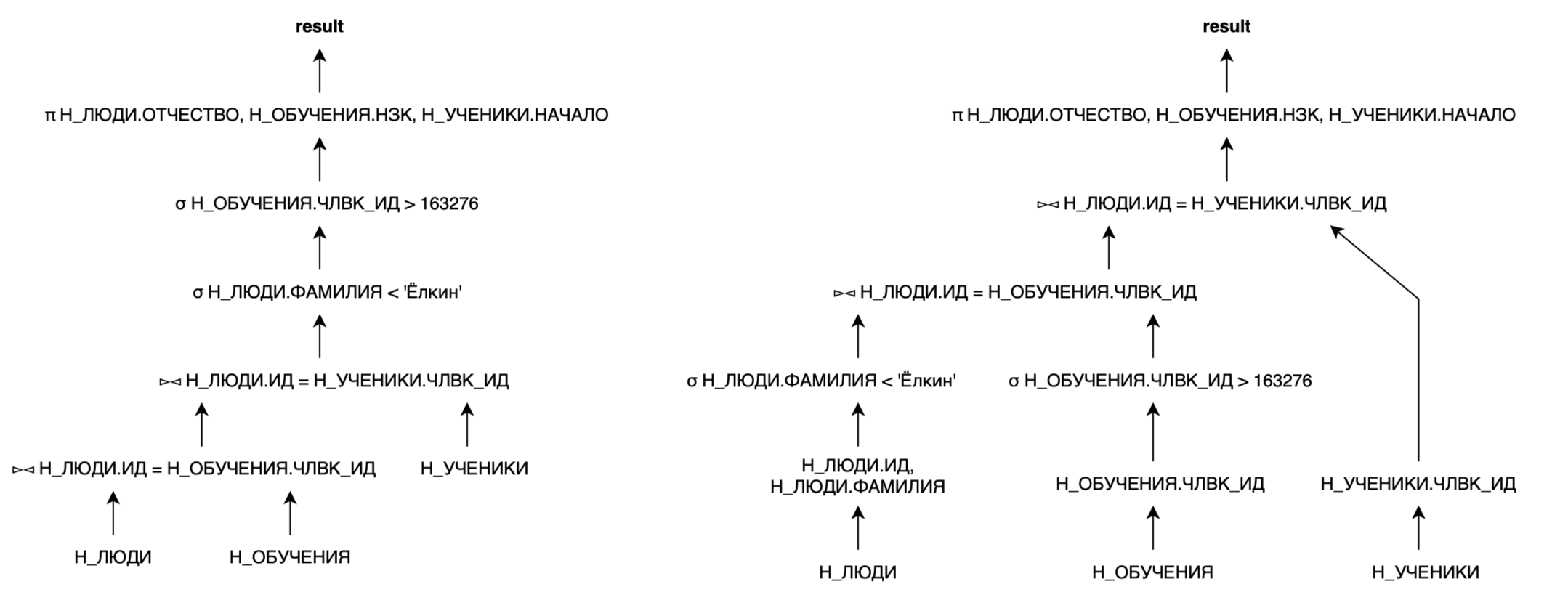
FROM Н\_ЛЮДИ

LEFT JOIN Н\_ОБУЧЕНИЯ ON Н\_ЛЮДИ.ИД = Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД

LEFT JOIN Н\_УЧЕНИКИ ON Н\_ЛЮДИ.ИД = Н\_УЧЕНИКИ.ЧЛВК\_ИД

WHERE (Н\_ЛЮДИ.ФАМИЛИЯ < 'Ёлкин' AND Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД > 163276);

*Планы выполнения:*



Второй план является оптимальным. Из-за того, что выборка происходит на более ранних этапах, идет соединение только нужных атрибутов, и размер промежуточных данных меньше.

*Индексы:*

CREATE INDEX ON "Н\_ЛЮДИ" USING HASH ("ИД");

CREATE INDEX ON "Н\_ЛЮДИ" USING BTREE ("ФАМИЛИЯ");

CREATE INDEX ON "Н\_ОБУЧЕНИЯ" USING BTREE ("ЧЛВК\_ИД");

Выборка происходит с использованием операторов сравнения, поэтому оптимально использование BTREE. Соединение таблиц с использованием ИД – оптимально использование HASH.

*Explain Analyze:*

EXPLAIN ANALYZE

SELECT Н\_ЛЮДИ.ОТЧЕСТВО, Н\_ОБУЧЕНИЯ.НЗК, Н\_УЧЕНИКИ.НАЧАЛО

FROM Н\_ЛЮДИ

LEFT JOIN Н\_ОБУЧЕНИЯ ON Н\_ЛЮДИ.ИД = Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД

LEFT JOIN Н\_УЧЕНИКИ ON Н\_ЛЮДИ.ИД = Н\_УЧЕНИКИ.ЧЛВК\_ИД

WHERE (Н\_ЛЮДИ.ФАМИЛИЯ < 'Ёлкин' AND Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД > 163276);

Nested Loop Left Join (cost=0.85..75.35 rows=9 width=34) (actual time=0.028..0.030 rows=1 loops=1)

-> Nested Loop (cost=0.56..71.08 rows=2 width=30) (actual time=0.022..0.022 rows=1 loops=1)

-> Index Scan using "ОБУЧ\_ЧЛВК\_FK\_I" on "Н\_ОБУЧЕНИЯ" (cost=0.28..21.25 rows=6 width=10) (actual time=0.007..0.008 rows=1 loops=1)

Index Cond: ("ЧЛВК\_ИД" > 163276)

-> Index Scan using "ЧЛВК\_PK" on "Н\_ЛЮДИ" (cost=0.28..8.30 rows=1 width=24) (actual time=0.009..0.010 rows=1 loops=1)

Index Cond: ("ИД" = "Н\_ОБУЧЕНИЯ"."ЧЛВК\_ИД")

Filter: (("ФАМИЛИЯ")::text < 'Ёлкин'::text)

-> Index Scan using "УЧЕН\_ОБУЧ\_FK\_I" on "Н\_УЧЕНИКИ" (cost=0.29..2.09 rows=5 width=12) (actual time=0.005..0.006 rows=1 loops=1)

Index Cond: ("ЧЛВК\_ИД" = "Н\_ЛЮДИ"."ИД")

Planning Time: 0.568 ms

Execution Time: 0.088 ms

# Заключение

В ходе выполнения данной лабораторной работы мне удалось разобраться индексами и обработкой СУБД запросов.